



**android**

# LẬP TRÌNH ANDROID 2

---

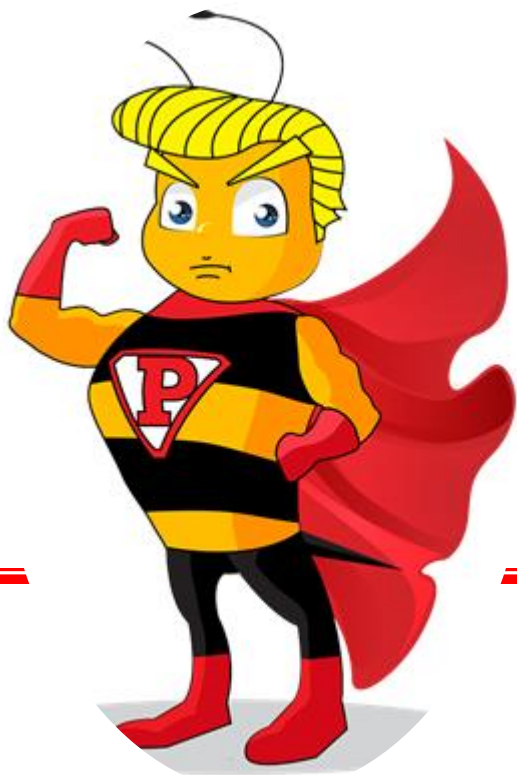
ALERTDIALOG

- ☐ Tổng quan AlertDialog
- ☐ Sử dụng AlertDialog
- ☐ Custom AlertDialog

# MỤC TIÊU

- ◎ TỔNG QUAN ALERTDIALOG
- ◎ SỬ DỤNG ALERTDIALOG
- ◎ CUSTOM ALERTDIALOG



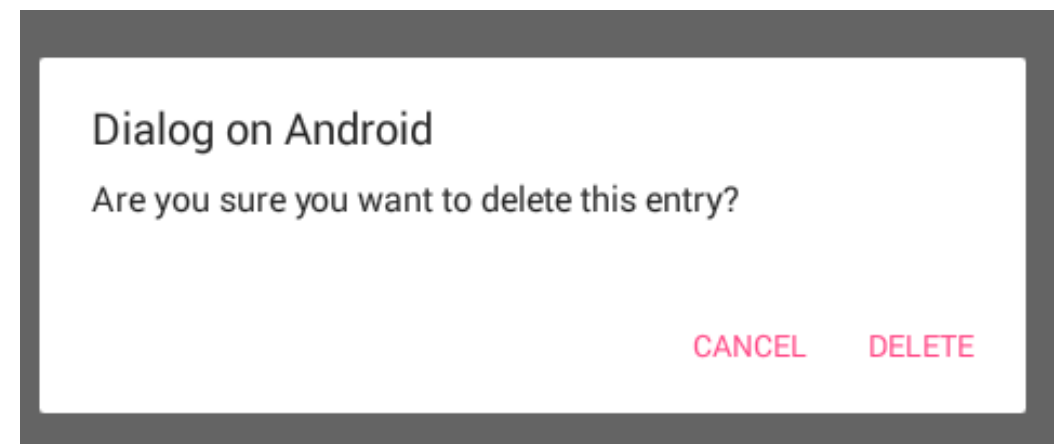
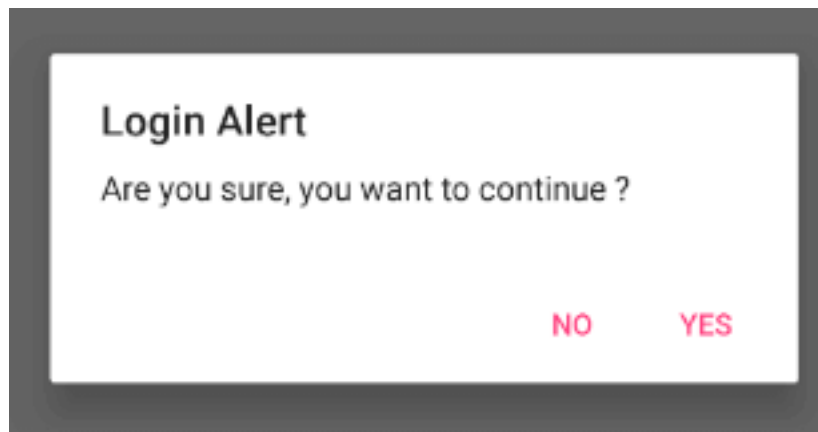


# GIỚI THIỆU ALERTDIALOG

---

...

❑ **Dialog** có thể coi là một hộp thoại mà người dùng có thể tương tác trực tiếp được. Ví dụ khi các bạn muốn xóa một tập tin quan trọng hay muốn thoát một chương trình nào đấy thì việc hiển thị một thông báo để người dùng chắc chắn về hành vi của mình là rất quan trọng.



❑ **AlertDialog** là lớp con của dialog

```
java.lang.Object
└─ android.app.Dialog
    └─ android.app.AlertDialog
```

❑ **AlertDialog** gồm có 3 vùng:

- ❑ Tiêu đề (title)
- ❑ Nội dung (content)
- ❑ Các button hành động.



- ❑ Để tạo Alert Dialog thường sử dụng thành phần **AlertDialog.Builder**
- ❑ **AlertDialog.Builder** được sử dụng để tạo ra một giao diện là một hộp thoại cảnh báo trong Android. Thường thì chúng ta phải thiết lập các vùng: Tiêu đề, nội dung, image, button và xử lý sự kiện cho button.

```
AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);
```



```
AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);
```



❑ Các phương thức thường dùng trong Alert Dialog

❑ **setTitle(CharSequence title)** – Phương thức này thường dùng để thiết lập tiêu đề cho Alert Dialog. ***Phương thức này tùy chọn.***

```
//Thiết lập tiêu đề cho alert dialog  
AlertDialogBuilder.setTitle("Thông báo");
```

❑ Các phương thức thường dùng trong Alert Dialog

❑ **setIcon(Drawable icon)** – Phương thức này dùng để thêm icon trước tiêu đề. Icon phải được lưu trong thư mục drawable.

```
// Icon Of Alert Dialog  
alertDialogBuilder.setIcon(R.drawable.ic_question);
```

- ❑ Các phương thức thường dùng trong Alert Dialog
  - ❑ **setMessage(CharSequence message)** – Phương thức này dùng hiển thị nội dung trong Alert Dialog. ***Nội dung này bắt buộc phải có.***

```
// Setting Alert Dialog Message  
alertDialogBuilder.setMessage("Bạn có thích Android không?");
```

- ❑ Các phương thức thường dùng trong Alert Dialog
  - ❑ **setCancelable(boolean cancelable)** – Phương thức này có 2 giá trị **true/false**.
    - ❑ Nếu thiết lập **false** thì khi show dialog lên người dùng click ra bên ngoài dialog thì nó vẫn không bị mất
    - ❑ Nếu set **true** thì sẽ mất khi click vào bất kì đâu ngoài dialog.

```
AlertDialogBuilder.setCancelable(false);
```

## ❑ Các phương thức thường dùng trong Alert Dialog

### ❑ **setPositiveButton(CharSequence text, DialogInterface.OnClickListener listener)**

Phương thức này thiết lập một Button Positive (Yes/Agree/...)

```
alertDialogBuilder.setPositiveButton(text: "Yes", new DialogInterface.OnClickListener() {  
    @Override  
    public void onClick(DialogInterface dialog, int which) {  
        //...  
    }  
});
```

❑ Các phương thức thường dùng trong Alert Dialog

❑ **setNegativeButton(CharSequence text, DialogInterface.OnClickListener listener)**

Phương thức này thiết lập một Button Negative (No/Không/...)

```
AlertDialogBuilder.setNegativeButton( text: "No", new DialogInterface.OnClickListener() {  
    @Override  
    public void onClick(DialogInterface dialog, int which) {  
        //...  
    }  
});
```

## ❑ Các phương thức thường dùng trong Alert Dialog

### ❑ `setNeutralButton(CharSequence text, DialogInterface.OnClickListener listener)`

Phương thức này đơn giản là thêm một Button mới, với button này chúng ta có thể xử lý sự kiện cho nó.

*Ví dụ 2 phương thức trên bạn đã thiết lập 2 Button "**Yes**" và "**No**", trong phương thức này chúng ta thiết lập Button "**Cancel**".*

```
AlertDialogBuilder.setNeutralButton(text: "Cancel", new DialogInterface.OnClickListener() {  
    @Override  
    public void onClick(DialogInterface dialog, int which) {  
        //...  
    }  
});
```



# TẠO ALERTDIALOG HOÀN CHỈNH

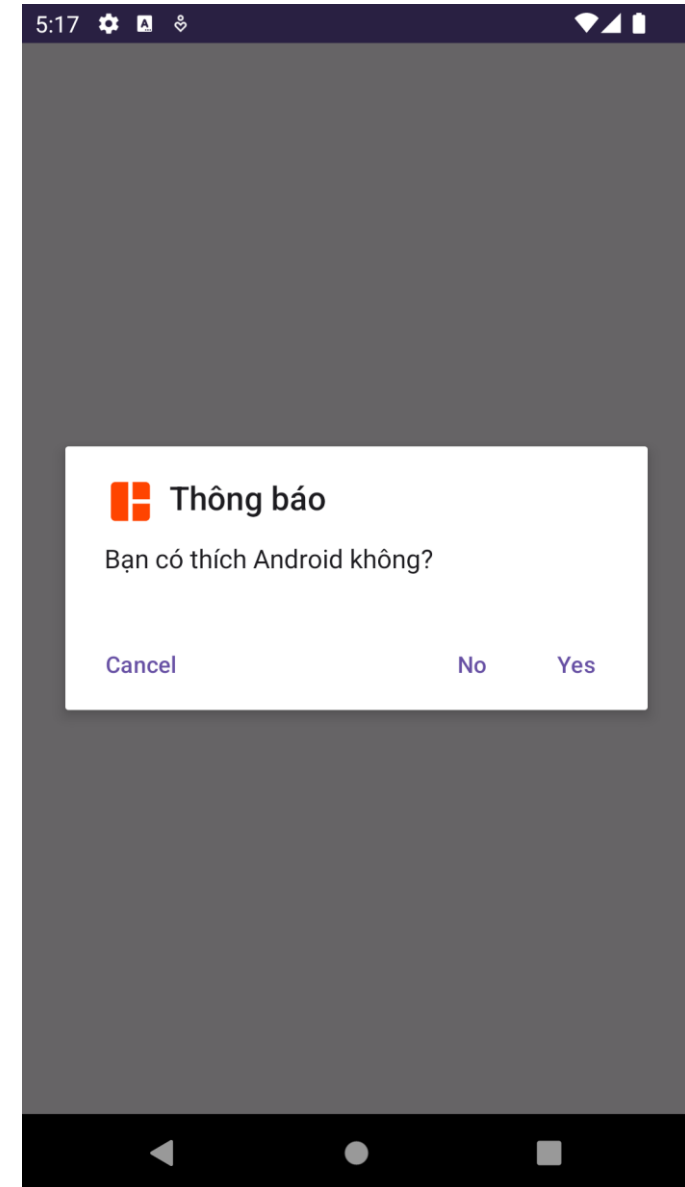
```
AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder( context: this);
alertDialogBuilder.setTitle("Thông báo");
alertDialogBuilder.setIcon(R.drawable.ic_question);
alertDialogBuilder.setMessage("Bạn có thích Android không?");
alertDialogBuilder.setCancelable(false);

alertDialogBuilder.setPositiveButton( text: "Yes", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        Toast.makeText( context: MainActivity.this, text: "Mình rất thích", Toast.LENGTH_SHORT).show();
    }
});

alertDialogBuilder.setNegativeButton( text: "No", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        Toast.makeText( context: MainActivity.this, text: "Mặc dù chọn KHÔNG nhưng vẫn thích", Toast.LENGTH_SHORT).show();
    }
});

alertDialogBuilder.setNeutralButton( text: "Cancel", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        Toast.makeText( context: MainActivity.this, text: "Mặc dù chọn HỦY nhưng vẫn thích", Toast.LENGTH_SHORT).show();
    }
});

AlertDialog alertDialog = alertDialogBuilder.create();
alertDialog.show();
```



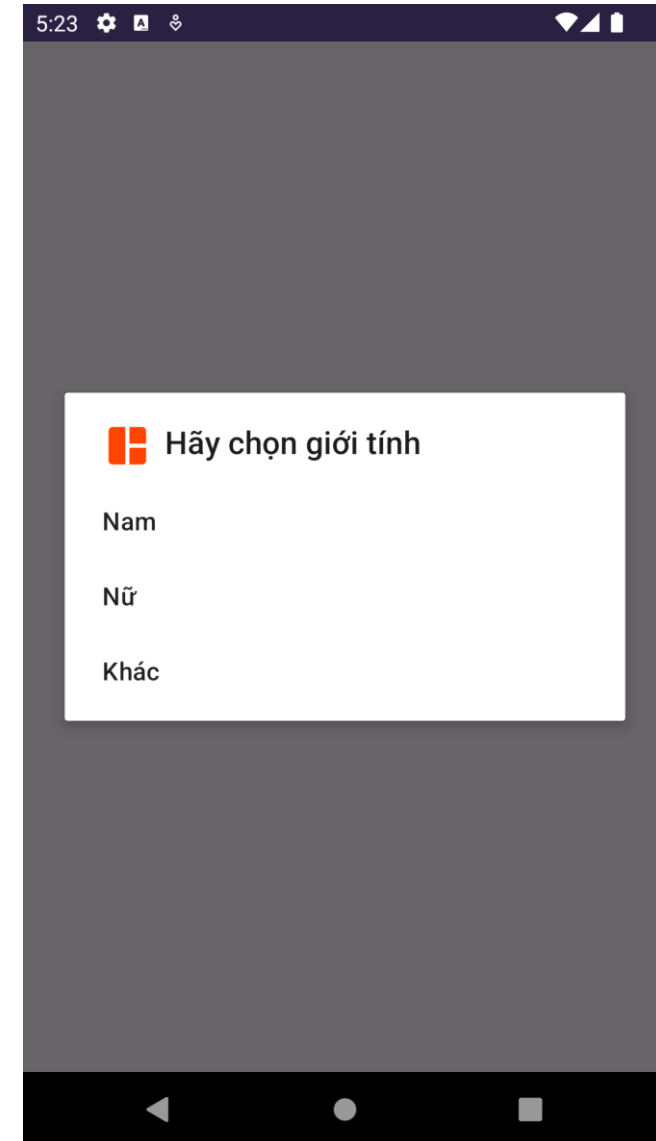
## AlertDialog dạng List

```
String[] gioiTinh = new String[]{"Nam", "Nữ", "Khác"};
```

```
AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(context: this);  
alertDialogBuilder.setTitle("Hãy chọn giới tính");  
alertDialogBuilder.setIcon(R.drawable.ic_question);
```

```
alertDialogBuilder.setItems(gioiTinh, new DialogInterface.OnClickListener() {  
    @Override  
    public void onClick(DialogInterface dialog, int which) {  
        Toast.makeText(context: MainActivity.this, text: "Giới tính: " + gioiTinh[which], Toast.LENGTH_SHORT).show();  
    }  
});
```

```
AlertDialog alertDialog = alertDialogBuilder.create();  
alertDialog.show();
```



## AlertDialog dạng Single Choice

```
String[] gioiTinh = new String[]{"Nam", "Nữ", "Khác"};

AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(context: this);
alertDialogBuilder.setTitle("Hãy chọn giới tính");
alertDialogBuilder.setIcon(R.drawable.ic_question);

alertDialogBuilder.setSingleChoiceItems(gioiTinh, checkedItem: 0, new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        Toast.makeText(context: MainActivity.this, text: "Giới tính: " + gioiTinh[which], Toast.LENGTH_SHORT).show();
    }
});

AlertDialog alertDialog = alertDialogBuilder.create();
alertDialog.show();
```



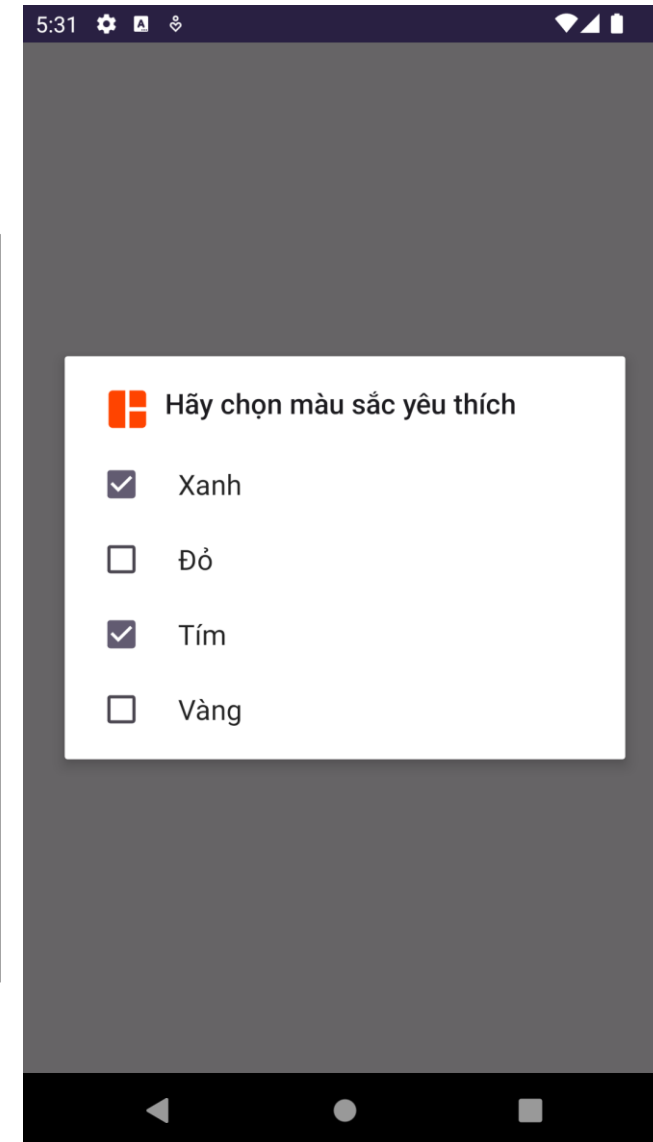
## AlertDialog dạng Multiple Choice

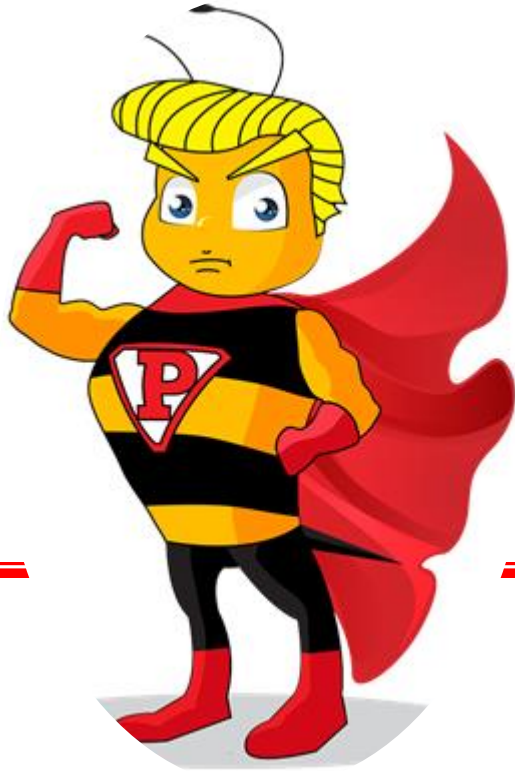
```
String[] mauSac = new String[]{"Xanh", "Đỏ", "Tím", "Vàng"};

AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(context: this);
alertDialogBuilder.setTitle("Hãy chọn màu sắc yêu thích");
alertDialogBuilder.setIcon(R.drawable.ic_question);

alertDialogBuilder.setMultiChoiceItems(mauSac, checkedItems: null, new DialogInterface.OnMultiChoiceClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which, boolean isChecked) {
        if (isChecked)
            Toast.makeText(context: MainActivity.this, text: "Bạn chọn màu " + mauSac[which], Toast.LENGTH_SHORT).show();
        else
            Toast.makeText(context: MainActivity.this, text: "Bạn bỏ chọn màu " + mauSac[which], Toast.LENGTH_SHORT).show();
    }
});

AlertDialog alertDialog = alertDialogBuilder.create();
alertDialog.show();
```





---

# CUSTOM ALERTDIALOG

---

...

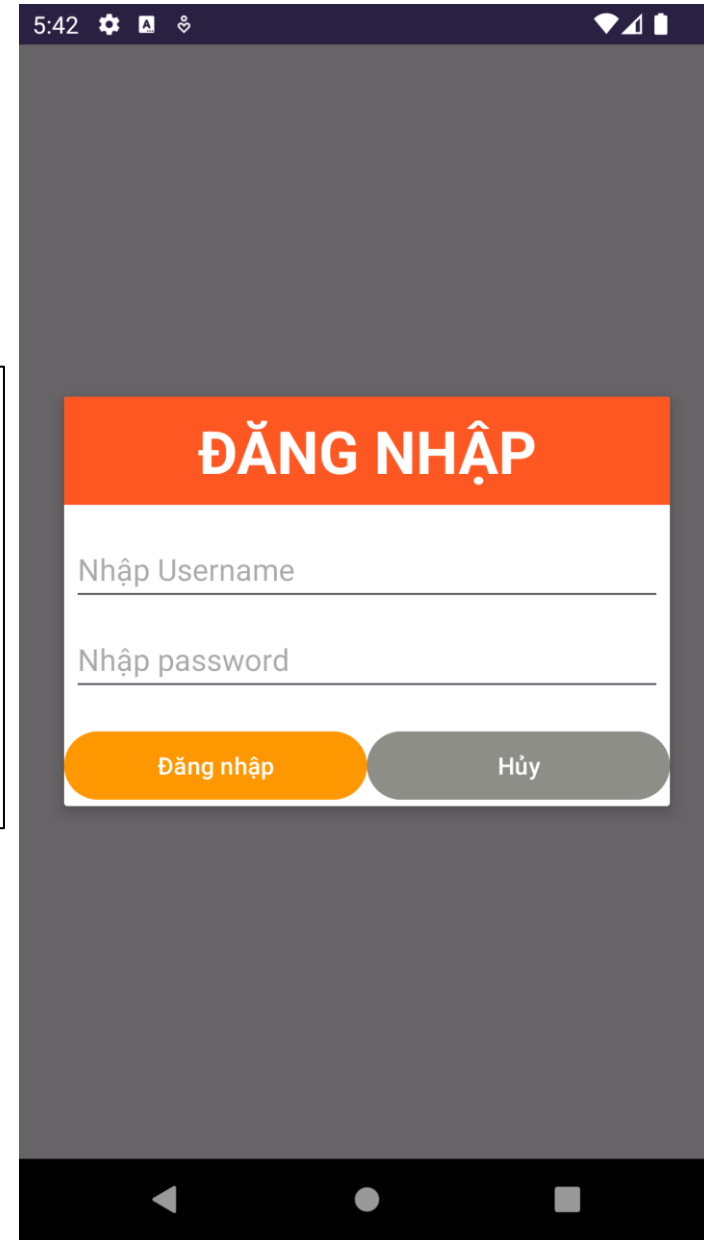
## ❑ Tạo một layout cho dialog

The image displays two side-by-side representations of a login dialog layout. The left representation is a realistic UI mockup with a white background and rounded corners. It features an orange header bar with the text 'ĐĂNG NHẬP' in white. Below the header, there are two input fields: 'Nhập Username' and 'Nhập password'. At the bottom, there are two buttons: an orange 'Đăng nhập' button and a grey 'Hủy' button. The right representation is a schematic diagram with a dark blue background. It shows a 'TextView' at the top, followed by two input fields labeled 'username' and 'password'. Below these are two buttons labeled 'Button'. The diagram uses dashed lines to indicate the layout structure.

## ❑ Tạo dialog, gắn layout và gán view

```
AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(context: this);  
LayoutInflater inflater = getLayoutInflater();  
View view = inflater.inflate(R.layout.dialog_login, root: null);  
alertDialogBuilder.setView(view);
```

```
AlertDialog alertDialog = alertDialogBuilder.create();  
alertDialog.show();
```



## □ Ảnh xạ và xử lý chức năng trong layout

```
AlertDialog alertDialog = alertDialogBuilder.create();

EditText edtUser = view.findViewById(R.id.edtUser);
EditText edtPass = view.findViewById(R.id.edtPass);
Button btnLogin = view.findViewById(R.id.btnLogin);
Button btnCancel = view.findViewById(R.id.btnCancel);

btnLogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String user = edtUser.getText().toString();
        String pass = edtPass.getText().toString();

        if (user.equals("poly") && pass.equals("android@2")) {
            Toast.makeText(context: MainActivity.this, text: "Đăng nhập thành công", Toast.LENGTH_SHORT).show();
            alertDialog.dismiss();
        } else {
            Toast.makeText(context: MainActivity.this, text: "Đăng nhập thất bại", Toast.LENGTH_SHORT).show();
        }
    }
});

btnCancel.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        alertDialog.dismiss();
    }
});

alertDialog.show();
```





**FPT** Education

FPT POLYTECHNIC

**Thank you**